

IN THE CLAIMS

Please cancel claims 2, 8, 12, 15, 18, and 21 without prejudice.

1. (Currently amended) A method for providing a plurality of client applications access to data in a distributed file system including a plurality of storage elements coupled to a plurality of read servers and a write server, comprising:

distinguishing between read requests and write requests from the client applications;
transmitting read requests from the client applications to the plurality of read servers;
transmitting write requests from the client applications to the write server, each write request including a replacement data element to replace a current data element at a current storage location on one of the storage elements;

reading file data by the read servers and returning the file data to the client applications in response to read requests; and

in response to each write request, writing data by the write server in response to write requests from the client applications;

writing the replacement data element to a new location in one of the storage elements while leaving the current data element accessible for read requests processed by the read servers;

locking a file meta-data element that references the current storage location;

updating the file meta-data element to reference the new location having the replacement data element; and

unlocking the file meta-data element after updating.

2. (Canceled)

3. (Currently amended) The method of claim 1 ~~2~~, further comprising:

allocating a new storage area at the new location prior to writing the replacement data;
deallocating an old storage area at the current storage location while the file meta-data element is locked.

4. (Original) The method of claim 1, further comprising balancing between the read servers a processing load associated with the read requests.

5. (Original) The method of claim 4, further comprising distributing read requests to the read servers in a round-robin distribution.

6. (Original) The method of claim 1, further comprising:
transmitting read and write requests to the read servers; and
transmitting write requests received at the read servers to the write servers.

7. (Currently amended) A system for providing a plurality of client applications access to data in a distributed file system including a plurality of storage elements, comprising:

a plurality of read servers coupled to the client applications and to the distributed file system, each read server configured to read file data from the distributed file system and return the file data to the client applications in response to read requests from the client applications;

a write server coupled to the client applications and to the distributed file system, the write server configured to write data to the distributed file system in response to write requests from the client applications, each write request including a replacement data element to replace a current data element at a current storage location on one of the storage elements;

means for distinguishing between read requests and write requests and transmitting read requests from the client applications to the plurality of read servers and transmitting write requests from the client applications to the write server[.];and

a data consistency control module hosted on the write server, the control module configured and arranged to write the replacement data element to a new location in one of the storage elements while leaving the current data element accessible to read requests processed by the read servers, lock a file meta-data element that references the current storage location while updating the file meta-data element to reference the new location having the replacement data element, and unlock the file meta-data element after updating.

8. (Canceled)

9. (Original) The method of claim 7, further comprising means for balancing between the read servers a processing load associated with the read requests.

10. (Original) The method of claim 9, further comprising means for distributing read requests to the read servers in a round-robin distribution.

11. (Currently amended) A system for providing a plurality of client applications access to data in a distributed file system including a plurality of storage elements, comprising:

a plurality of read servers coupled to the client applications and to the distributed file system, each read server configured to read file data from the distributed file system and return the file data to the client applications in response to read requests from the client applications and configured to transmit write requests to a write server;

the write server coupled to the client applications and to the distributed file system, the write server configured to write data to the distributed file system in response to write requests from the client applications, each write request including a replacement data element to replace a current data element at a current storage location on one of the storage elements;

a load balancer coupled to the client applications and to the plurality of read servers, the load balancer configured and arranged to selectively transmit read requests and write requests from the client applications to the plurality of read servers as a function of respective levels of processing loads of the read servers, wherein each of the read servers is further configured and arranged to distinguish between read requests and write requests and transmit write requests to the write server[.]; and

a data consistency control module hosted on the write server, the control module configured and arranged to write the replacement data element to a new location in one of the storage elements while leaving the current data element accessible to read requests processed by the read servers, lock a file meta-data element that references the current storage location while updating the file meta-data element to reference the new location having the replacement data element, and unlock the file meta-data element after updating.

12. (Canceled)

13. (Currently amended) The system of claim 11 ~~12~~, wherein the load balancer is configured and arranged to distribute the read and write requests to the read servers in round-robin distribution.

14. (Currently amended) A system for providing a plurality of client applications access to data in a distributed file system including a plurality of storage elements, comprising:

a plurality of read servers coupled to the client applications and to the distributed file system, each read server configured to read file data from the distributed file system and return the file data to the client applications in response to read requests from the client applications and configured to transmit write requests to a write server;

the write server coupled to the client applications and to the distributed file system, the write server configured to write data to the distributed file system in response to write requests from the client applications, each write request including a replacement data element to replace a current data element at a current storage location on one of the storage elements;

a load balancer coupled to the client applications, to the plurality of read servers, and to the write server, the load balancer configured and arranged to distinguish between read requests and write requests and selectively transmit read requests from the client applications to the plurality of read servers as a function of respective levels of processing loads of the read servers and configured and arranged to transmit write requests to the write server[.]; and

a data consistency control module hosted on the write server, the control module configured and arranged to write the replacement data element to a new location in one of the storage elements while leaving the current data element accessible to read requests processed by the read servers, lock a file meta-data element that references the current storage location while updating the file meta-data element to reference the new location having the replacement data element, and unlock the file meta-data element after updating.

15. (Canceled)

16. (Original) The system of claim 14, wherein the load balancer is configured and arranged to distribute the read requests to the read servers in round-robin distribution.

17. (Currently amended) A system for providing a plurality of client applications access to data in a distributed file system including a plurality of storage elements, comprising:

a plurality of read servers coupled to the client applications and to the distributed file system, each read server configured to read file data from the distributed file system and

return the file data to the client applications in response to read requests from the client applications and configured to transmit write requests to a write server;

the write server coupled to the client applications and to the distributed file system, the write server configured to write data to the distributed file system in response to write requests from the client applications, each write request including a replacement data element to replace a current data element at a current storage location on one of the storage elements;

a plurality of load balancers, each load balancer coupled to a respective one of the client applications, to the plurality of read servers, and to the write server, each load balancer configured and arranged to distinguish between read requests and write requests and selectively transmit read requests from a coupled client application to the plurality of read servers as a function of respective levels of processing loads of the read servers and configured and arranged to transmit write requests to the write server[.]; and

a data consistency control module hosted on the write server, the control module configured and arranged to write the replacement data element to a new location in one of the storage elements while leaving the current data element accessible to read requests processed by the read servers, lock a file meta-data element that references the current storage location while updating the file meta-data element to reference the new location having the replacement data element, and unlock the file meta-data element after updating.

18. (Canceled)

19. (Original) The system of claim 17, wherein each of the load balancers is configured and arranged to distribute the read requests to the read servers in round-robin distribution.

20. (Currently amended) A system for providing a plurality of client applications access to data in a distributed file system including a plurality of storage elements, comprising:

a plurality of read servers coupled to the client applications and to the distributed file system, each read server configured to read file data from the distributed file system and return the file data to the client applications in response to read requests from the client applications and configured to transmit write requests to a write server;

a write server coupled to the client applications and to the distributed file system, the write server configured to write data to the distributed file system in response to write requests from the client applications, each write request including a replacement data

element to replace a current data element at a current storage location on one of the storage elements;

a plurality of load balancers, each load balancer coupled to the plurality of read servers, each load balancer configured and arranged to selectively transmit read requests and write requests from client applications to the plurality of read servers as a function of respective levels of processing loads of the read servers, wherein each of the read servers is further configured and arranged to distinguish between read requests and write requests and transmit write requests to the write server[.]; and

a data consistency control module hosted on the write server, the control module configured and arranged to write the replacement data element to a new location in one of the storage elements while leaving the current data element accessible to read requests processed by the read servers, lock a file meta-data element that references the current storage location while updating the file meta-data element to reference the new location having the replacement data element, and unlock the file meta-data element after updating.

21. (Canceled)

22. (original) The system of claim 20, wherein each of the load balancers is configured and arranged to distribute the read requests and the write requests to the read servers in round-robin distribution.

23. (New) A method for providing a plurality of client applications access to data in a distributed file system, comprising:

identifying read requests and write requests from the client applications;
selecting in response to each read request, one of a plurality of read servers;
sending each read request to the selected one of the read servers;
sending each write request to a write server, each write request including a first data value to replace a current data value at a first storage location on one of a plurality of storage elements coupled to the read servers and to the write server;

reading from a storage element coupled to a read server, in response to a read request received at the read server, file data and returning the file data to a requesting client application; and

in response to each write request received at the write server,

writing the first data value to a second location in one of the storage elements while leaving the current data value accessible for read requests processed by the read servers, wherein the first location is different from the second location;

locking a file meta-data element that references the first storage location;

updating, while the file meta-data element is locked, the file meta-data element to reference the second location; and

unlocking the file meta-data element after updating.

24. (New) The method of claim 23, further comprising:

allocating a new storage area at the second location prior to writing the first data value;

deallocating a storage area at the first storage location while the file meta-data element is locked.

25. (New) The method of claim 23, further comprising balancing between the read servers a processing load associated with the read requests.

26. (New) The method of claim 25, further comprising distributing read requests to the read servers in a round-robin distribution.

27. (New) The method of claim 23, further comprising:

transmitting read and write requests from the client applications to the read servers;
and

transmitting write requests received at the read servers to the write servers.

28. (New) An apparatus for providing a plurality of client applications access to data in a distributed file system, comprising:

means for identifying read requests and write requests from the client applications;

means for selecting in response to each read request, one of a plurality of read servers;

means for sending each read request to the selected one of the read servers;

means for sending each write request to a write server, each write request including a first data value to replace a current data value at a first storage location on one of a plurality of storage elements coupled to the read servers and to the write server;

means for reading from a storage element coupled to a read server, in response to a read request received at the read server, file data and returning the file data to a requesting client application; and

means, responsive to each write request received at the write server, for writing the first data value to a second location in one of the storage elements while leaving the current data value accessible for read requests processed by the read servers, wherein the first location is different from the second location, for locking a file meta-data element that references the first storage location, for updating, while the file meta-data element is locked, the file meta-data element to reference the second location, and for unlocking the file meta-data element after updating.

29. (New) The apparatus of claim 28, further comprising means for balancing between the read servers a processing load associated with the read requests.